Welcome to E-XFL.COM

Embedded - Microcontrollers - Application Specific: Tailored Solutions for Precision and Performance

Embedded - Microcontrollers - Application Specific represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering enhanced performance, efficiency, and functionality to meet the demands of specialized applications.

What Are Embedded - Microcontrollers - Application Specific?

Application-specific microcontrollers are engineered to

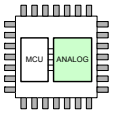
Details	
Product Status	Obsolete
Applications	Automotive
Core Processor	S12
Program Memory Type	FLASH (32KB)
Controller Series	HCS12
RAM Size	2K x 8
Interface	LIN, SCI
Number of I/O	9
Voltage - Supply	2.25V ~ 5.5V
Operating Temperature	-40°C ~ 105°C
Mounting Type	Surface Mount
Package / Case	48-LQFP Exposed Pad
Supplier Device Package	48-HLQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mm912f634cv2aer2

4 Functional Description and Application Information

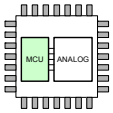
4.1 Introduction

This chapter describes the MM912F634 dual die device functions on a block by block base. To distinguish between the module location being the MCU die or the analog die, the following symbols are shown on all module cover pages:

The documented module is physically located on the Analog die. This applies to Section 4.2, “MM912F634 - Analog Die Overview" through Section 4.25, “MM912F634 - Analog Die Trimming".



The documented module is physically located on the Microcontroller die. This applies to Section 4.26, “MM912F634 - MCU Die Overview" through Section 4.38, “Serial Peripheral Interface (S12SPIV4)".



Sections concerning both dies or the complete device does not have a specific indication.

4.1.1 Device Register Maps

Table 47 shows the device register memory map overview for the 32 kByte MCU die (MC9S12I32).

NOTE

Reserved register space shown in Table 47 is not allocated to any module. This register space is reserved for future use, and shows as grayed areas in tables throughout this document. Writing to these locations has no effect. Read access to these locations returns zero.

Table 47. Device Register Memory Map Overview

Address	Module	Size (Bytes)
0x0000–0x0007	PIM (port integration module)	8
0x0008–0x0019	Reserved	18
0x001A–0x001B	Part ID register	2
0x001C–0x001E	Reserved	3
0x001F	INT (interrupt module)	1
0x0020–0x002F	DBG (debug module)	16
0x0030-0x0033	MMC (memory map control)	4
0x0034–0x003B	CRG (clock and reset generator)	8
0x003C–0x003D	RTI (real time interrupt)	2
0x003E–0x003F	COP (computer operating properly)	2
0x0040–0x00D7	Reserved	152
0x00D8–0x00DF	D2DI (die 2 die initiator)	8
0x00E0–0x00E7	Reserved	8
0x00E8-0x00EF	SPI (serial peripheral interface)	8
0x00F0–0x00FF	Reserved	16
0x0100–0x0113	FTSR control registers	20
0x0114–0x011F	Reserved	12
0x0120–0x0123	PIM (port integration module)	4
0x0124–0x01FF	Reserved	220

0x8C-0 x97	ADRx (hi)	R	adrx 9	adrx 8	adrx 7	adrx 6	adrx 5	adrx 4	adrx 3	adrx 2
	ADC Data Result Register x	W								
	ADRx (lo)	R	adrx 1	adrx 0	0	0	0	0	0	0
	ADC Data Result Register x	W								

4.2.3.2.3 Functional Considerations

For the not available Lx inputs, the following functions are limited:

- No Wake-up feature / Cyclic Sense
- No Digital Input
- No Analog Input and conversion via ADC

4.3 Modes of Operation

The MM912F634 analog die offers three main operating modes: Normal (Run), Stop, and Sleep. In Normal mode, the device is active and is operating under normal application conditions. In Stop mode, the voltage regulator operates with limited current capability, the external load is expected to be reduced while in Stop mode. In Sleep mode both voltage regulators are turned off (V_{DD} = V_{DDX} = 0 V).

Wake-up from Stop mode is indicated by an interrupt signal. Wake-up from Sleep mode changes the MM912F634 analog die into reset mode while the voltage regulator is turned back on.

The selection of the different modes is controlled by the Mode Control Register (MCR).

Figure 16 describes how transitions are done between the different operating modes.

Table 109. PWM Register Summary (continued)

Name / Offset[89]		7	6	5	4	3	2	1	0
0x69 PWMDTY1	R W	Bit 7	6	5	4	3	2	1	Bit 0

Note:
89. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

4.13.3.1 PWM Control Register (PWMCTL)

Table 110. PWM Control Register (PWMCTL)

Offset[90] 0x60 Access: User read/write

	7	6	5	4	3	2	1	0
R W	CAE1	CAE0	PCLK1	PCLK0	PPOL1	PPOL0	PWME1	PWME0
Reset	0	0	0	0	0	0	0	0

Note:
90. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 111. PWMCTL - Register Field Descriptions

Field	Description
7–6 CAE[1:0]	Center Aligned Output Modes on Channels 1–0 0 Channels 1–0 operate in left aligned output mode. 1 Channels 1–0 operate in center aligned output mode.
5 PCLK1	Pulse Width Channel 1 Clock Select 0 Clock B is the clock source for PWM channel 1. 1 Clock SB is the clock source for PWM channel 1.
4 PCLK0	Pulse Width Channel 0 Clock Select 0 Clock A is the clock source for PWM channel 0. 1 Clock SA is the clock source for PWM channel 0.
3–2 PPOL[1:0]	Pulse Width Channel 1–0 Polarity Bits 0 PWM channel 1–0 outputs are low at the beginning of the period, then go high when the duty count is reached. 1 PWM channel 1–0 outputs are high at the beginning of the period, then go low when the duty count is reached.
1-0 PWME[1:0]	Pulse Width Channel 1–0 Enable 0 Pulse width channel 1–0 is disabled. 1 Pulse width channel 1–0 is enabled. The pulse modulated signal becomes available at PWM, output bit 1 when its clock source begins its next cycle.

4.13.3.1.1 PWM Enable (PWMEx)

NOTE

The first PWM cycle after enabling the channel can be irregular. If both PWM channels are disabled (PWME1–0 = 0), the prescaler counter shuts off for power savings.

Each PWM channel has an enable bit (PWMEx) to start its waveform output. When any of the PWMEx bits are set (PWMEx = 1), the associated PWM output is enabled immediately. However, the actual PWM waveform is not available on the associated PWM output until its clock source begins its next cycle, due to the synchronization of PWMEx and the clock source.

4.13.3.1.2 PWM Polarity (PPOLx)

NOTE

PPOLx register bits can be written anytime. If the polarity changes while a PWM signal is being generated, a truncated or stretched pulse can occur during the transition

After receiving the stop bit into the receive shifter, and provided the receive data register is not already full, the data character is transferred to the receive data register and the receive data register full (RDRF) status flag is set. If RDRF was already set indicating the receive data register (buffer) was already full, the overrun (OR) status flag is set and the new data is lost. Because the SCI receiver is double-buffered, the program has one full character time after RDRF is set before the data in the receive data buffer must be read to avoid a receiver overrun.

When a program detects the receiving data register is full (RDRF = 1), it gets the data from the receive data register by reading SCID. The RDRF flag is cleared automatically by a 2-step sequence which is normally satisfied in the course of the user's program handling receive data. Refer to Section 4.15.3.4, "Interrupts and Status Flags" for more details about flag clearing.

4.15.3.3.1 Data Sampling Technique

The SCI receiver uses a 16× baud rate clock for sampling. The receiver starts by taking logic level samples at 16 times the baud rate to search for a falling edge on the RxD serial data input pin. A falling edge is defined as a logic 0 sample after three consecutive logic 1 samples. The 16× baud rate clock is used to divide the bit time into 16 segments labeled RT1 through RT16. When a falling edge is located, three more samples are taken at RT3, RT5, and RT7 to make sure this was a real start bit and not merely noise. If at least two of these three samples are 0, the receiver assumes it is synchronized to a receive character.

The receiver then samples each bit time, including the start and stop bits, at RT8, RT9, and RT10 to determine the logic level for this bit. The logic level is interpreted to be of the majority of the samples taken during the bit time. In the case of the start bit, the bit is assumed to be 0 if at least two of the samples at RT3, RT5, and RT7 are 0 even if one or all of the samples taken at RT8, RT9, and RT10 are 1s. If any sample in any bit time (including the start and stop bits) in a character frame fails to agree with the logic level for this bit, the noise flag (NF) is set when the received character is transferred to the receive data buffer.

The falling edge detection logic continuously looks for falling edges, and if an edge is detected, the sample clock is resynchronized to bit times. This improves the reliability of the receiver in the presence of noise or mismatched baud rates. It does not improve worst case analysis because some characters do not have any extra falling edges anywhere in the character frame.

In the case of a framing error, provided the received character was not a break character, the sampling logic searching for a falling edge is filled with three logic 1 samples so a new start bit can be detected almost immediately.

In the case of a framing error, the receiver is inhibited from receiving any new characters until the framing error flag is cleared. The receive shift register continues to function, but a complete character cannot transfer to the receive data buffer if FE is still set.

4.15.3.3.2 Receiver Wake-up Operation

Receiver wake-up is a hardware mechanism allowing an SCI receiver to ignore the characters in a message intended for a different SCI receiver. In such a system, all receivers evaluate the first character(s) of each message, and as soon as they determine the message is intended for a different receiver, they write logic 1 to the receiver wake up (RWU) control bit in SCIC2. When RWU bit is set, the status flags associated with the receiver (with the exception of the idle bit, IDLE, when RWUID bit is set) are inhibited from setting, thus eliminating the software overhead for handling the unimportant message characters. At the end of a message, or at the beginning of the next message, all receivers automatically force RWU to 0 so all receivers wake up in time to look at the first character(s) of the next message.

4.15.3.3.2.1 Idle-line Wake-up

When WAKE = 0, the receiver is configured for idle-line wake-up. In this mode, RWU is cleared automatically when the receiver detects a full character time of the idle-line level. The M control bit selects 8-bit or 9-bit data mode which determines how many bit times of idle are needed to constitute a full character time (10 or 11 bit times because of the start and stop bits).

When RWU is one and RWUID is zero, the idle condition waking up the receiver does not set the IDLE flag. The receiver wakes up and waits for the first data character of the next message which sets the RDRF flag and generate an interrupt if enabled. When RWUID is one, any idle condition sets the IDLE flag and generates an interrupt if enabled, regardless of whether RWU is zero or one.

The idle-line type (ILT) control bit selects one of two ways to detect an idle line. When ILT = 0, the idle bit counter starts after the start bit so the stop bit and any logic 1s at the end of a character count toward the full character time of idle. When ILT = 1, the idle bit counter does not start until after a stop bit time, so the idle detection is not affected by the data in the last character of the previous message.

9-bit data mode typically is used in conjunction with parity to allow eight bits of data plus the parity in the ninth bit. Or it is used with address-mark wake-up so the ninth data bit can serve as the wake-up bit. In custom protocols, the ninth bit can also serve as a software-controlled marker.

4.15.3.5.2 Stop Mode Operation

During all stop modes, clocks to the SCI module are halted. In stop1 and stop2 modes, all SCI register data is lost and must be re-initialized upon recovery from these two stop modes. No SCI module registers are affected in stop3 mode.

The receive input active edge detect circuit is still active in stop3 mode, but not in stop2. An active edge on the receive input brings the CPU out of stop3 mode if the interrupt is not masked (RXEDGIE = 1).

Note that because the clocks are halted, the SCI module resumes operation upon exit from stop (only in stop3 mode). Software should ensure stop mode is not entered while there is a character being transmitted out of or received into the SCI module.

4.15.3.5.3 Loop Mode

When LOOPS = 1, the RSRC bit in the same register chooses between loop mode (RSRC = 0) or single-wire mode (RSRC = 1). Loop mode is sometimes used to check software, independent of connections in the external system, to help isolate system problems. In this mode, the transmitter output is internally connected to the receiver input and the RxD pin is not used by the SCI, so it reverts to a general purpose port I/O pin.

4.15.3.5.4 Single-wire Operation

When LOOPS = 1, the RSRC bit in the same register chooses between loop mode (RSRC = 0) or single-wire mode (RSRC = 1). Single-wire mode is used to implement a half-duplex serial connection. The receiver is internally connected to the transmitter output and to the TxD pin. The RxD pin is not used and reverts to a general purpose port I/O pin.

In single-wire mode, the TXDIR bit in SCIC3 controls the direction of serial data on the TxD pin. When TXDIR = 0, the TxD pin is an input to the SCI receiver and the transmitter is temporarily disconnected from the TxD pin so an external device can send serial data to the receiver. When TXDIR = 1, the TxD pin is an output driven by the transmitter. In single-wire mode, the internal loop back connection from the transmitter to the receiver causes the receiver to receive characters sent out by the transmitter.

4.16 High Voltage Inputs - Lx

Six High Voltage capable inputs are implemented with the following features:

- Digital Input Capable
- Analog Input Capable with selectable voltage divider.
- Wake-up Capable during Low Power mode. See Section 4.8, “Wake-up / Cyclic Sense".

When used as analog inputs to sense voltages outside the module a series resistor must be used on the used input. When a Lx input is not selected in the analog multiplexer, the voltage divider is disconnected from this input. When a Lx input is selected in the analog multiplexer, it is disconnected in low power mode if configured as Wake-up input. Unused Lx pins are recommended to be connected to GND to improve EMC behavior.

4.16.1 Register Definition

4.16.1.1 Lx Status Register (LXR)

Table 140. Lx Status Register (LXR)

Offset[109] 0x08 Access: User read

	7	6	5	4	3	2	1	0
R	0	0	L5	L4	L3	L2	L1	L0
W								

Note:
109. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

4.17.4.2 Port B Configuration Register 2 (PTBC2)

Table 147. Port B Configuration Register 2 (PTBC2)

Offset[112] 0x21 Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	PWMCS	PWMEN	SERMOD	
W								
Reset	0	0	0	0	0	0	0	0

Note:
112. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 148. PTBC2 - Register Field Descriptions

Field	Description
3 PWMCS	PWM Channel Select PTB2. See Section 4.13, “PWM Control Module (PWM8B2C)". 0 - PWM Channel 0 selected as PWM Channel for PTB2 1 - PWM Channel 1 selected as PWM Channel for PTB2
2 PWMEN	PWM Enable for PTB2. See Section 4.13, “PWM Control Module (PWM8B2C)". 0 - PWM disabled on PTB2 1 - PWM enabled on PTB2 (Channel as selected with PWMCS)
1-0 SERMOD	Serial Mode Select for PTB0 and PTB1. See Figure 35 for details. 00 - Mode 0, SCI internally connected the LIN Physical Layer Interface. PTB0 and PTB1 are Digital I/Os 01 - Mode 1, SCI connected to PTB0 and PTB1 (external SCI mode) 10 - Mode 2, LIN Physical Layer Interface connected to PTB0 and PTB1 (external LIN mode) 11 - Mode 3, SCI internally connected the LIN Physical Layer Interface and PTB0 and PTB1 are connected both as outputs (Observe mode)

4.17.4.3 Port B Data Register (PTB)

Table 149. Port B Data Register (PTB)

Offset[113] 0x22 Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	PTB2	PTB1	PTB0
W								
Reset	0	0	0	0	0	0	0	0

Note:
113. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 150. PTB - Register Field Descriptions

Field	Description
2-0 PTB[2-0]	Port B general purpose input/output data — Data Register If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered and synchronized pin input state is read.

Table 176. Timer Clock Selection

PR2	PR1	PR0	Timer Clock
0	0	0	D2D Clock / 1
0	0	1	D2D Clock / 2
0	1	0	D2D Clock / 4
0	1	1	D2D Clock / 8
1	0	0	D2D Clock / 16
1	0	1	D2D Clock / 32
1	1	0	D2D Clock / 64
1	1	1	D2D Clock / 128

4.18.3.3.12 Main Timer Interrupt Flag 1 (TFLG1)

NOTE

These flags are set when an input capture or output compare event occurs. Flag set on a particular channel is cleared by writing a one to the corresponding CnF bit. Writing a zero to CnF bit has no effect on its status. When TFFCA bit in TSCR register is set, a read from an input capture or a write into an output compare channel causes the corresponding channel flag CnF to be cleared.

Table 177. Main Timer Interrupt Flag 1 (TFLG1)

Offset[129] 0xCC — Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	C3F	C2F	C1F	C0F
W								
Reset	0	0	0	0	0	0	0	0

Note:
129. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 178. TFLG1 - Register Field Descriptions

Field	Description
3-0 C[3:0]F	Input Capture/Output Compare Channel Flag. 1 = Input Capture or Output Compare event occurred 0 = No event (Input Capture or Output Compare event) occurred.

4.18.3.3.13 Main Timer Interrupt Flag 2 (TFLG2)

NOTE

The TFLG2 register indicates when an interrupt has occurred. Writing a one to the TOF bit clears it. Any access to TCNT clears TOF bit of TFLG2 register if the TFFCA bit in TSCR register is set.

Table 179. Main Timer Interrupt Flag 2 (TFLG2)

Offset[130] 0xCD — Access: User read/write

	7	6	5	4	3	2	1	0
R	TOF	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Note:
130. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 180. TFLG2 - Register Field Descriptions

Field	Description
7 TOF	Timer Overflow Flag 1 = Indicates an Interrupt has occurred (Set when 16-bit free-running timer counter overflows from $FFFF to $0000) 0 = Flag indicates an Interrupt has not occurred.

4.18.3.3.14 Timer Input Capture/Output Compare Registers (TC3 - TC0)

NOTE

TRead anytime. Write anytime for output compare function. Writes to these registers have no effect during input capture.

Depending on the TIOS bit for the corresponding channel, these registers are used to latch the value of the free-running counter when a defined transition is sensed by the corresponding input capture edge detector or to trigger an output action for output compare.

Read/Write access in byte mode for high byte should takes place before low byte otherwise it gives a different result.

Table 181. Timer Input Capture/Output Compare Register 0 (TC0)

Offset[131] 0xCE, 0xCF — Access: User read(anytime)/write (special mode)

	15	14	13	12	11	10	9	8
R W	tc0_15	tc0_14	tc0_13	tc0_12	tc0_11	tc0_10	tc0_9	tc0_8
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R W	tc0_7	tc0_6	tc0_5	tc0_4	tc0_3	tc0_2	tc0_1	tc0_0
Reset	0	0	0	0	0	0	0	0

Note:
131. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 182. Timer Input Capture/Output Compare Register 1(TC1)

Offset[132] 0xD0, 0xD1 — Access: User read(anytime)/write (special mode)

	15	14	13	12	11	10	9	8
R W	tc1_15	tc1_14	tc1_13	tc1_12	tc1_11	tc1_10	tc1_9	tc1_8
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R W	tc1_7	tc1_6	tc1_5	tc1_4	tc1_3	tc1_2	tc1_1	tc1_0
Reset	0	0	0	0	0	0	0	0

Note:
132. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

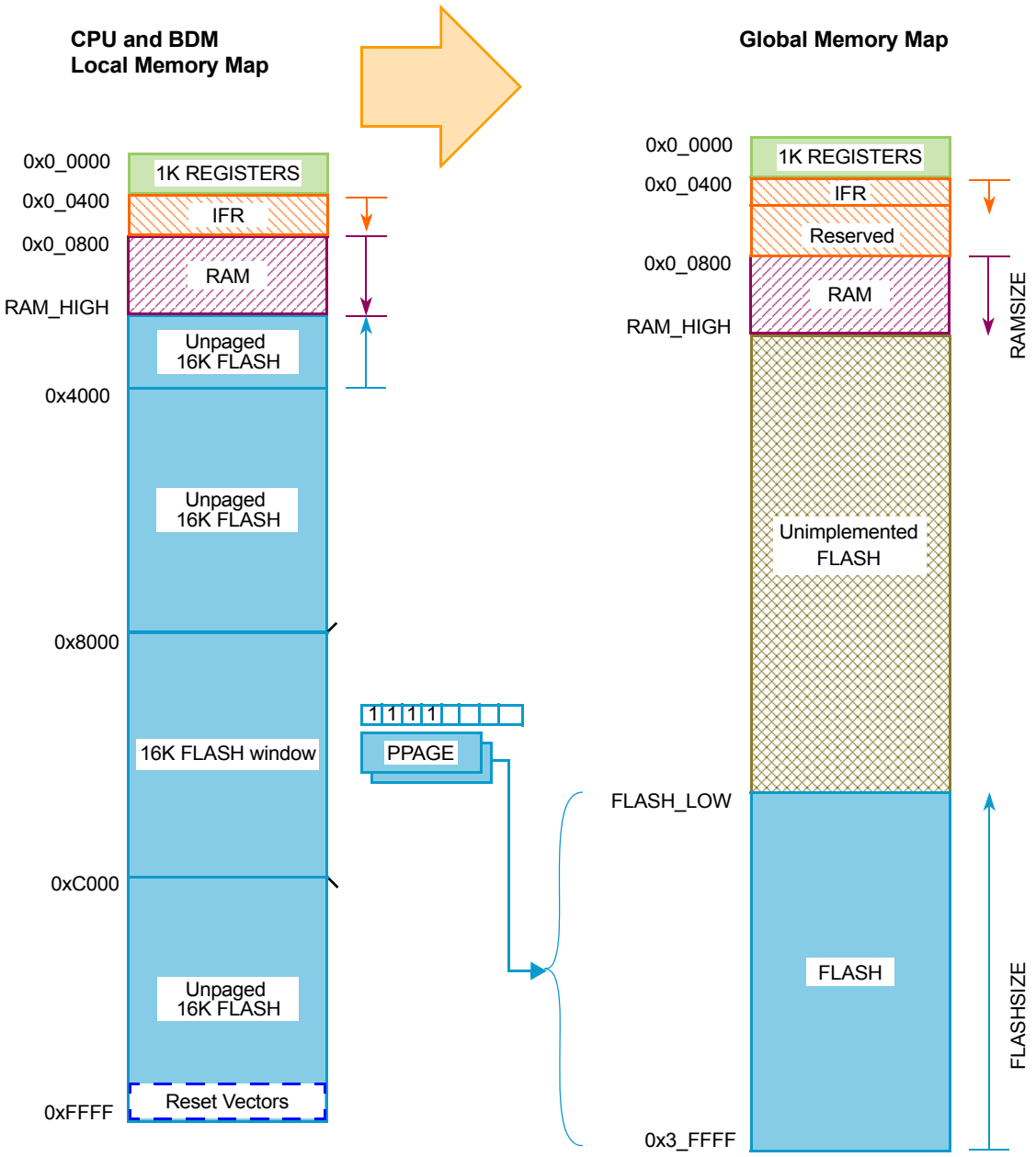


Figure 45. MC9S12I32 Global Address Mapping

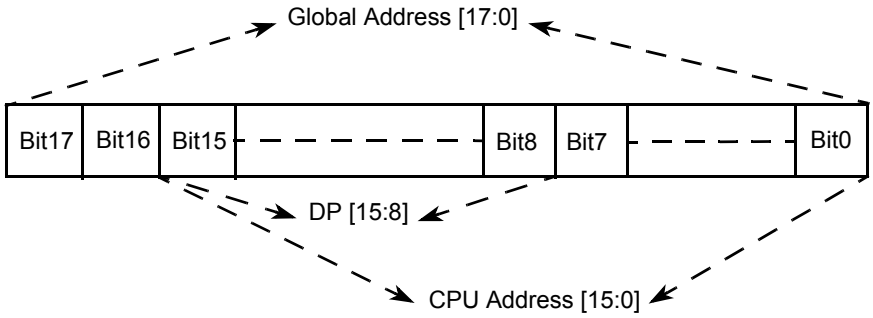


Figure 50. Direct Address Mapping

Example 1. This Example Demonstrates Usage of the Direct Addressing Mode

```
MOVB    #0x80,DIRECT    ;Set DIRECT register to 0x80. Write once only.
                        ;Global data accesses to the range 0xXX_80XX can be direct.
                        ;Logical data accesses to the range 0x80XX are direct.

LDY     <00             ;Load the Y index register from 0x8000 (direct access).
                        ;< operator forces direct access on some assemblers but in
                        ;many cases assemblers are “direct page aware” and can
                        ;automatically select direct mode.
```

4.28.2.2.3 Mode Register (MODE)

Table 245. Mode Register (MODE)

Address: 0x0032

	7	6	5	4	3	2	1	0
R	MODC	0	0	0	0	0	0	0
W								
Reset	MODC[172]	0	0	0	0	0	0	0

Note:
172. External signal (see Table 239).

Read: Anytime.
Write: Only if a transition is allowed (see Figure 51).

The MODC bit of the MODE register is used to establish the MCU operating mode.

Table 246. MODE Field Descriptions

Field	Description
7 MODC	**Mode Select Bit** — This bit controls the current operating mode during RESET high (inactive). The external mode pin MODC determines the operating mode during RESET low (active). The state of the pin is registered into the respective register bit after the RESET signal goes inactive (see Figure 51). Write restrictions exist to disallow transitions between certain modes. Figure 51 illustrates all allowed mode changes. Attempting non authorized transitions do not change the MODE bit, but it blocks further writes to the register bit except in special modes. Changes of operating modes are not allowed when the device is secured, but it blocks further writes to the register bit except in special modes.

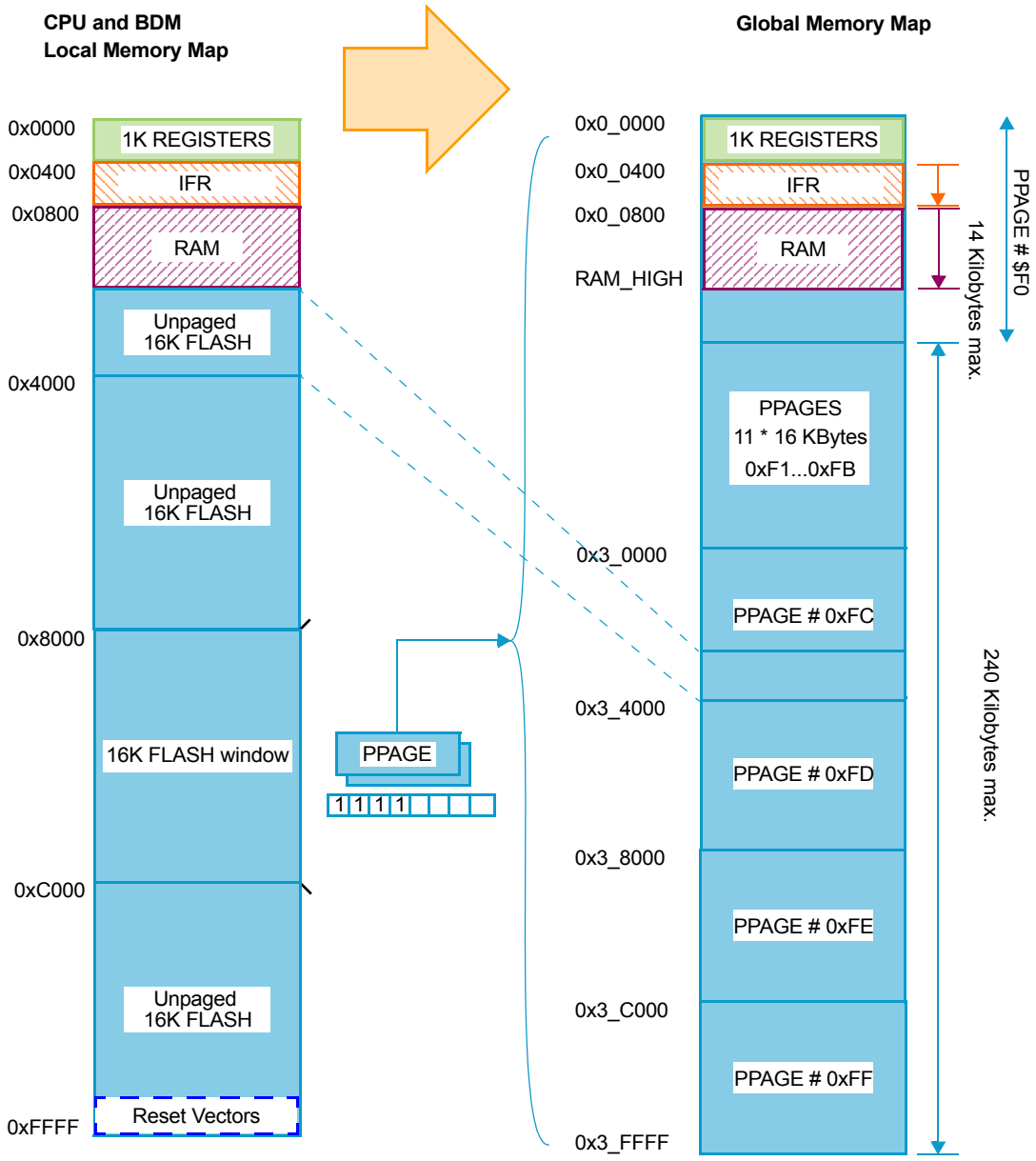


Figure 54. Local to Global Address Mapping

Table 250. IVBR Field Descriptions

Field	Description
7–0 IVB_ADDR[7:0]	**Interrupt Vector Base Address Bits** — These bits represent the upper byte of all vector addresses. Out of reset these bits are set to 0xFF (i.e., vectors are located at 0xFF80–0xFFFE) to ensure compatibility to HCS12. **Note:** A system reset initializes the interrupt vector base register with “0xFF” before it is used to determine the reset vector address. Therefore, changing the IVBR has no effect on the location of the three reset vectors (0xFFFA–0xFFFE). **Note:** If the BDM is active (i.e., the CPU is in the process of executing BDM firmware code), the contents of IVBR are ignored and the upper byte of the vector address is fixed as “0xFF”. This is done to enable handling of all non-maskable interrupts in the BDM firmware.

4.29.4 Functional Description

The 9S12I32PIMV1 module processes all exception requests to be serviced by the CPU module. These exceptions include interrupt vector requests and reset vector requests. Each of these exception types and their overall priority level is discussed in the subsections below.

4.29.4.1 S12S Exception Requests

The CPU handles both reset requests and interrupt requests. A priority decoder is used to evaluate the priority of pending interrupt requests.

4.29.4.2 Interrupt Prioritization

NOTE

All non I bit maskable interrupt requests always have higher priority than the I bit maskable interrupt requests. If the X bit in the CCR is cleared, it is possible to interrupt an I bit maskable interrupt by an X bit maskable interrupt. It is possible to nest non maskable interrupt requests, e.g., by nesting SWI or TRAP calls.

Care must be taken to ensure all interrupt requests remain active until the system begins execution of the applicable service routine. Otherwise, the exception request may not get processed at all or the result may be a spurious interrupt request (vector at address (vector base + 0x0080)).

The 9S12I32PIMV1 module contains a priority decoder to determine the priority for all interrupt requests pending for the CPU. If more than one interrupt request is pending, the interrupt request with the higher vector address wins the prioritization.

The following conditions must be met for an I bit maskable interrupt request to be processed.

1. The local interrupt enabled bit in the peripheral module must be set.
2. The I bit in the condition code register (CCR) of the CPU must be cleared.
3. There is no SWI, TRAP, or X bit maskable request pending.

Since an interrupt vector is only supplied at the time when the CPU requests it, it is possible a higher priority interrupt request could override the original interrupt request causing the CPU to request the vector. In this case, the CPU receives the highest priority vector and the system processes this interrupt request first, before the original interrupt request is processed.

If the interrupt source is unknown (for example, in the case where an interrupt request becomes inactive after the interrupt has been recognized, but prior to the CPU vector request), the vector address supplied to the CPU defaults to the spurious interrupt vector.

4.29.4.3 Reset Exception Requests

The 9S12I32PIMV1 module supports three system reset exception request types (refer to CRG for details):

1. Pin reset, power-on reset or illegal address reset
2. Clock monitor reset request
3. COP watchdog reset request

4.29.4.4 Exception Priority

The priority (from highest to lowest) and address of all exception vectors issued by the 9S12I32PIMV1 module upon request by the CPU is shown in Table 251.

Table 251. Exception Vector Map and Priority

Vector Address[175]	Source
0xFFFE	Pin reset, power-on reset, illegal address reset
0xFFFC	Clock monitor reset
0xFFFA	COP watchdog reset
(Vector base + 0x00F8)	Unimplemented opcode trap
(Vector base + 0x00F6)	Software interrupt instruction (SWI) or BDM vector request
(Vector base + 0x00F4)	X bit maskable interrupt request (D2DI error interrupt)
(Vector base + 0x00F2)	D2DI interrupt request
(Vector base + 0x00F0–0x0082)	Device specific I bit maskable interrupt sources (priority determined by the low byte of the vector address, in descending order)
(Vector base + 0x0080)	Spurious interrupt

Note:
175. 16 bits vector address based

4.29.5 Initialization/Application Information

4.29.5.1 Initialization

After system reset, software should:

1. Initialize the interrupt vector base register if the interrupt vector table is not located at the default location (0xFF80–0xFFF9).
2. Enable I bit maskable interrupts by clearing the I bit in the CCR.
3. Enable the X bit maskable interrupt by clearing the X bit in the CCR.

4.29.5.2 Interrupt Nesting

The interrupt request scheme makes it possible to nest I bit maskable interrupt requests handled by the CPU.

- I bit maskable interrupt requests can be interrupted by an interrupt request with a higher priority.

I bit maskable interrupt requests cannot be interrupted by other I bit maskable interrupt requests, per default. In order to make an interrupt service routine (ISR) interruptible, the ISR must explicitly clear the I bit in the CCR (CLI). After clearing the I bit, other I bit maskable interrupt requests can interrupt the current ISR.

An ISR of an interruptible I bit maskable interrupt request could basically look like this:

1. Service interrupt, e.g., clear interrupt flags, copy data, etc.
2. Clear I bit in the CCR by executing the instruction CLI (thus allowing other I bit maskable interrupt requests)
3. Process data
4. Return from interrupt by executing the instruction RTI

4.30.2 External Signal Description

A single-wire interface pin called the background debug interface (BKGD) pin is used to communicate with the BDM system. During reset, this pin is a mode select input which selects between normal and special modes of operation. After reset, this pin becomes the dedicated serial interface pin for the background debug mode. The communication rate of this pin is based on the DCO clock or external reference clock depending on the configuration selected (refer to the S12S_CRG Block Guide for more details) which gets divided by five. Hence the BDM serial interface clock is always DCO clock divided by five after reset in to Special Single Chip mode which is about 6.4 MHz. After reset the BDM communication rate can be modified either via BDM command or CPU user code. When modifying the DCO clock make sure the communication rate is adapted accordingly and a communication timeout (BDM soft reset) has occurred.

4.30.3 Memory Map and Register Definition

4.30.3.1 Module Memory Map

Table 252 shows the BDM memory map when BDM is active.

Table 252. BDM Memory Map

Global Address	Module	Size (Bytes)
0x3_FF00–0x3_FF0B	BDM registers	12
0x3_FF0C–0x3_FF0E	BDM firmware ROM	3
0x3_FF0F	Family ID (part of BDM firmware ROM)	1
0x3_FF10–0x3_FFFF	BDM firmware ROM	240

4.30.3.2 Register Descriptions

A summary of the registers associated with the BDM is shown in Table 253. Registers are accessed by host-driven communications to the BDM hardware using READ_BD and WRITE_BD commands.

Table 253. BDM Register Summary

Global Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x3_FF00	Reserved	R	X	X	X	X	X	X	0	0
		W								
0x3_FF01	BDMSTS	R	ENBDM	BDMACT	0	SDV	TRACE	0	UNSEC	0
		W								
0x3_FF02	Reserved	R	X	X	X	X	X	X	X	X
		W								
0x3_FF03	Reserved	R	X	X	X	X	X	X	X	X
		W								
0x3_FF04	Reserved	R	X	X	X	X	X	X	X	X
		W								
0x3_FF05	Reserved	R	X	X	X	X	X	X	X	X
		W								
0x3_FF06	BDMCCR	R	CCR7	CCR6	CCR5	CCR4	CCR3	CCR2	CCR1	CCR0
		W								
0x3_FF07	Reserved	R	0	0	0	0	0	0	0	0
		W								

Figure 59 represents the BDM command structure. The command blocks illustrate a series of eight bit times starting with a falling edge. The bar across the top of the blocks indicates the BKGD line idles in the high state. The time for an 8-bit command is 8 × 16 target clock cycles. Target clock cycles are cycles measured using the target MCU's serial clock rate. See Section 4.30.4.6, "BDM Serial Interface" and Section 4.30.3.2.1, "BDM Status Register (BDMSTS)" for information on how serial clock rate is selected.

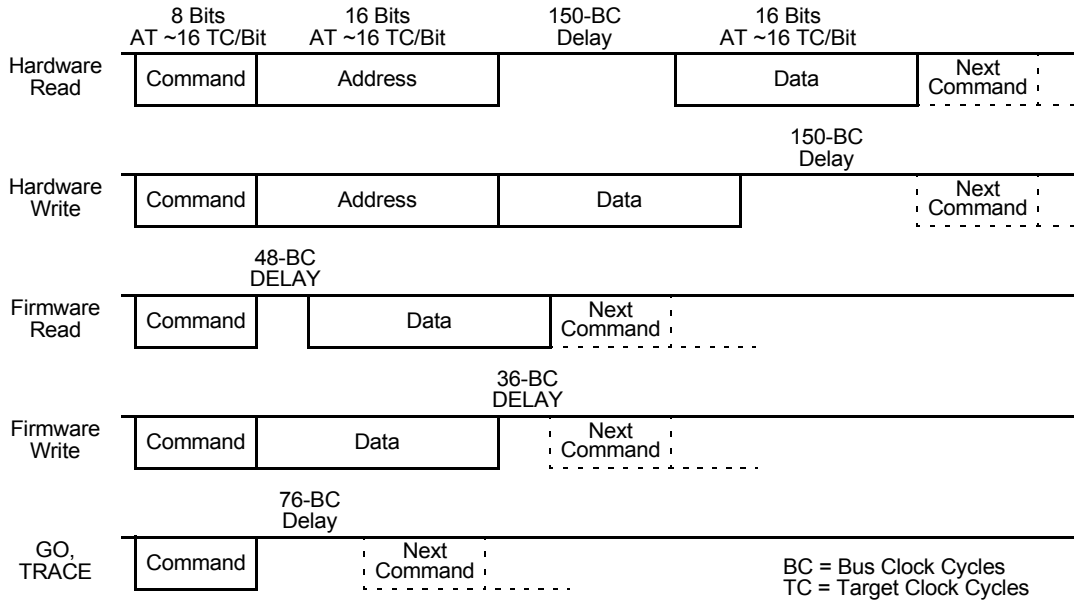


Figure 59. BDM Command Structure

4.30.4.6 BDM Serial Interface

The BDM communicates with external devices serially via the BKGD pin. During reset, this pin is a mode select input which selects between normal and special modes of operation. After reset, this pin becomes the dedicated serial interface pin for the BDM. The BDM serial interface is timed based on DCO clock or external reference clock depending on the configuration used (refer to the CRG Block Guide for more details), which gets divided by five. This clock is referred to as the target clock in the following explanation.

The BDM serial interface uses a clocking scheme in which the external host generates a falling edge on the BKGD pin to indicate the start of each bit time. This falling edge is sent for every bit whether data is transmitted or received. Data transfers the most significant bit (MSB) first at 16 target clock cycles per bit. The interface times out if 512 clock cycles occur between the falling edges from the host. The BKGD pin is a pseudo open-drain pin and has an weak on-chip active pull-up enabled at all times. It is assumed there is an external pull-up and drivers connected to BKGD do not typically drive the high level. Since R-C rise time could be unacceptably long, the target system and host provide brief driven high (speedup) pulses to drive BKGD to a logic 1. The source of this speedup pulse is the host for transmit cases and the target for receive cases.

The timing for a host-to-target is shown in Figure 60 and a target-to-host in Figure 61 and Figure 62. All four cases begin when the host drives the BKGD pin low to generate a falling edge. Since the host and target are operating from separate clocks, it can take the target system up to one full clock cycle to recognize this edge. The target measures delays from this perceived start of the bit time while the host measures delays from the point it actually drove BKGD low to start the bit up to one target clock cycle earlier. Synchronization between the host and target is established in this manner at the start of every bit time.

Figure 60 shows an external host transmitting a logic 1 and transmitting a logic 0 to the BKGD pin of a target system. The host is asynchronous to the target, so there is up to a one clock-cycle delay from the host-generated falling edge to where the target recognizes this edge as the beginning of the bit time. Ten target clock cycles later, the target senses the bit level on the BKGD pin. Internal glitch detect logic requires the pin be driven high no later than eight target clock cycles after the falling edge for a logic 1 transmission.

Since the host drives the high speedup pulses in these two cases, the rising edges look like digitally driven signals.

Each set of comparator registers can be accessed using the COMRV bits in the DBGC1 register. Unimplemented registers (e.g. Comparator B data bus and data bus masking) read as zero and cannot be written. The control register for comparator B differs from those of comparators A and C.

Table 291. Comparator Register Layout

0x0028	CONTROL	Read/Write	Comparators A,B and C
0x0029	ADDRESS HIGH	Read/Write	Comparators A,B and C
0x002A	ADDRESS MEDIUM	Read/Write	Comparators A,B and C
0x002B	ADDRESS LOW	Read/Write	Comparators A,B and C
0x002C	DATA HIGH COMPARATOR	Read/Write	Comparator A only
0x002D	DATA LOW COMPARATOR	Read/Write	Comparator A only
0x002E	DATA HIGH MASK	Read/Write	Comparator A only
0x002F	DATA LOW MASK	Read/Write	Comparator A only

4.31.3.2.8.1 Debug Comparator Control Register (DBGXCTL)

The contents of this register bits 7 and 6 differ depending upon which comparator registers are visible in the 8-byte window of the DBG module register address map.

Table 292. Debug Comparator Control Register DBGACTL (Comparator A)

Address: 0x0028

	7	6	5	4	3	2	1	0
R	0	NDB	TAG	BRK	RW	RWE	0	COMPE
W								
Reset	0	0	0	0	0	0	0	0

Table 293. Debug Comparator Control Register DBGBCTL (Comparator B)

Address: 0x0028

	7	6	5	4	3	2	1	0
R	SZE	SZ	TAG	BRK	RW	RWE	0	COMPE
W								
Reset	0	0	0	0	0	0	0	0

Table 294. Debug Comparator Control Register DBGCCTL (Comparator C)

Address: 0x0028

	7	6	5	4	3	2	1	0
R	0	0	TAG	BRK	RW	RWE	0	COMPE
W								
Reset	0	0	0	0	0	0	0	0

Read: DBGACTL if COMRV[1:0] = 00
DBGBCTL if COMRV[1:0] = 01
DBGCCTL if COMRV[1:0] = 10

4.34.7.2 Modification of Prescaler rate (RTIRT bits)

Applications which modify the Frequency Divider Rate by modifying the RTIRT bits (Prescaler rate) in the RTICTL register should follow below recommendations.

If the Frequency Divider Rate is set lower or equal to three the RTI interrupt service routine accesses the RTICTL register with in a timing window which is less or equal the synchronization delay. Hence the interrupt service routine accessing the RTICTL register to clear the RTIF bit is executed with such frequency, the RTIRT bits are permanently locked. Therefore the following sequence is recommended if RTIRT bits should be changed for a current selected Frequency Divider Rate of two or three:

- Access the RTICTL register to clear the RTI interrupt flag (RTIF bit) and disable the RTI interrupt (clear RTIE bit) by a single write access.
- Execute a software loop in which the RTICTL register is written to modify the RTIRT bits until the new Frequency Divider Rate is taken (read back value of RTIRT bits equals new value)
- Access RTICTL register to enable RTI interrupts again.

If the actual Frequency Divider Rate of the RTI is set to a rate higher than three the write access to clear the interrupt flag (RTIF bit) in the RTICTL register can be used to modify the RTIRT bits of the RTICTL register.

4.34.7.3 Modification of Modulus Down Counter Rate (RTICNT register)

Applications which frequently access the RTICNT register should follow below recommendations. If the RTICNT register is accessed with in a timing window which is less or equal the synchronization delay the following sequence is recommended:

- Access the RTICTL register to clear the RTI interrupt flag (RTIF bit) and disable the RTI interrupt (clear RTIE bit) by a single write access.
- Execute a software loop in which the RTICNT register is written to modify the rate until the new Frequency Divider Rate is taken (read back value of RTICNT bits equals new value)
- Access RTICTL register to enable RTI interrupts again.

If the RTICNT register is accessed in a timing window which is higher than the synchronization delay, only the RTICNT register needs to be written and wait until next time-out occurs.

4.35 Computer Operating Properly (S12SCOPV1)

4.35.1 Introduction

This section describes the functionality of the Computer Operating Properly module (COP), a sub-block of the HCS12S core platform.The COP (free running watchdog timer) enables the user to check that a program is running and sequencing properly. If the COP times out a system reset is initiated. Two types of COP operation are available: Window COP or Normal COP

When COP is enabled, sequential writes of $55 and $AA (in this order) are expected to the ARMCOP register during the selected timeout period. Once this is done, the COP timeout period restarts. If the program fails to do this the S12SCOP initiates a reset.

4.35.1.1 Overview

A block diagram of the COP is shown in Figure 79

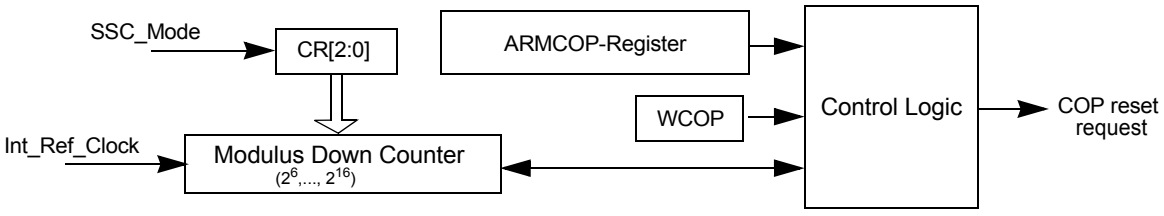


Figure 80. Block Diagram

4.36.4 Functional Description

4.36.4.1 Flash Command Operations

Flash command operations are used to execute program, erase, and erase verify algorithms described in this section. The program and erase algorithms are controlled by the Flash memory controller whose time base, FCLK, is derived from the bus clock via a programmable divider.

The next sections describe:

1. How to write the FCLKDIV register to set FCLK
2. Command write sequences to program, erase, and erase verify operations on the Flash memory
3. Valid Flash commands
4. Effects resulting from illegal Flash command write sequences or aborting Flash operations

4.36.4.1.1 Writing the FCLKDIV Register

NOTE

The values loaded into the FCLKDIV register are different from those loaded into the FCLKDIV register on prior S12 Flash modules, as they were based on the oscillator frequency.

Prior to issuing any Flash command after a reset, the user is required to write the FCLKDIV register to divide the bus clock down to within the 150 to 200 kHz range.

If it is defined:

- FCLK as the clock of the Flash timing control block
- INT(x) as taking the integer part of x (e.g. INT(4.323) = 4)

then FCLKDIV bits PRDIV8 and FDIV[5:0] are to be set as described in Figure 83.

For example, if the bus clock frequency is 20 MHz, FCLKDIV bits FDIV[5:0] should be set to 0x0C (001100), and bit PRDIV8 set to 1. The resulting FCLK frequency is then 192 kHz. In this case, the Flash program and erase algorithm timings are increased over the optimum target by:

$$(200 - 192)/200 = 4\% \qquad \textit{Eqn. 1}$$

CAUTION

Program and erase command execution time increases proportionally with the period of FCLK. Programming or erasing the Flash memory with FCLK < 150 kHz should be avoided. Setting FCLKDIV to a value such that FCLK < 150 kHz can destroy the Flash memory due to overstress. Setting FCLKDIV to a value such that FCLK > 200 kHz can result in incomplete programming or erasure of the Flash memory cells.

If the FCLKDIV register is written, the FDIVLD bit is set automatically. If the FDIVLD bit is 0, the FCLKDIV register has not been written since the last reset. If the FCLKDIV register has not been written to, the Flash command loaded during a command write sequence does not execute and the ACCERR flag in the FSTAT register sets.

4.37.3.2.4 D2DI Status Register 1 (D2DSTAT1)

This register holds the status of the external interrupt pin and an indicator about the D2DI transaction status.

Table 399. D2DI Status Register 1 (D2DSTAT1)

0x00DB — Access: User read

	7	6	5	4	3	2	1	0
R	D2DIF	D2DBSY	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Table 400. D2DSTAT1 Register Field Descriptions

Field	Description
7 D2DIF	**D2D Interrupt Flag** — This read-only flag reflects the status of the D2DINT Pin. The D2D interrupt flag can only be cleared by a target specific interrupt acknowledge sequence. 0 External Interrupt is negated 1 External Interrupt is asserted
6 D2DBSY	**D2D Initiator Busy** — This read-only status bit indicates a D2D transaction is ongoing. 0 D2D initiator idle. 1 D2D initiator transaction ongoing.
5:0	Reserved, should be masked to ensure compatibility with future versions of this interface.

4.37.3.2.5 D2DI Address Buffer Register (D2DADR)

This read-only register contains information about the ongoing D2D interface transaction. The register content is updated when a new transaction starts. In error cases the user can track back, which transaction failed.

Table 401. D2DI Address Buffer Register (D2DADR)

0x00DC / 0x00DD — Access: User read

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RWB	SZ8	0	NBLK	0	0	0	0	ADR[7:0]							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 402. D2DI Address Buffer Register Bit Descriptions

Field	Description
15 RWB	**Transaction Read-Write Direction** — This read-only bit reflects the direction of the transaction 0 Write Transaction 1 Read Transaction
14 SZ8	**Transaction Size** — This read-only bit reflects the data size of the transaction 0 16-bit transaction. 1 8-bit transaction.
13	Reserved, should be masked to ensure compatibility with future versions of this interface.
12 NBLK	**Transaction Mode** — This read-only bit reflects the mode of the transaction 0 Blocking transaction. 1 Non-blocking transaction.

- The SCK is output for the master mode and input for the slave mode.
- The $\overline{SS}$ is the input or output for the master mode, and it is always the input for the slave mode.
- The bidirectional mode does not affect SCK and $\overline{SS}$ functions.

4.38.4.6 Error Conditions

The SPI has one error condition:

- Mode fault error

4.38.4.6.1 Mode Fault Error

NOTE

If a mode fault error occurs and a received data byte is pending in the receive shift register, this data byte is lost.

If the $\overline{SS}$ input becomes low while the SPI is configured as a master, it indicates a system error where more than one master may be trying to drive the MOSI and SCK lines simultaneously. This condition is not permitted in normal operation. The MODF bit in the SPI status register is set automatically, provided the MODFEN bit is set.

In the special case where the SPI is in master mode and MODFEN bit is cleared, the $\overline{SS}$ pin is not used by the SPI. In this case, the mode fault error function is inhibited and MODF remains cleared. In case the SPI system is configured as a slave, the $\overline{SS}$ pin is a dedicated input pin. Mode fault error doesn't occur in slave mode.

If a mode fault error occurs, the SPI is switched to slave mode, with the exception that the slave output buffer is disabled. So SCK, MISO, and MOSI pins are forced to be high-impedance inputs, to avoid any possibility of conflict with another output driver. A transmission in progress is aborted and the SPI is forced into idle state.

If the mode fault error occurs in the bidirectional mode for a SPI system configured in master mode, output enable of the MOMI (MOSI in bidirectional mode) is cleared if it was set. No mode fault error occurs in the bidirectional mode for SPI system configured in slave mode.

The mode fault flag is cleared automatically by a read of the SPI status register (with MODF set), followed by a write to the SPI control register 1. If the mode fault flag is cleared, the SPI becomes a normal master or slave again.

4.38.4.7 Low Power Mode Options

4.38.4.7.1 SPI in Run Mode

In run mode with the SPI system enable (SPE) bit in the SPI control register clear, the SPI system is in a low-power, disabled state. SPI registers remain accessible, but clocks to the core of this module are disabled.

4.38.4.7.2 SPI in Wait Mode

NOTE

Care must be taken when expecting data from a master while the slave is in wait or stop mode. Even though the shift register continues to operate, the rest of the SPI is shut down (i.e., a SPIF interrupt is **not** generated until exiting stop or wait mode). Also, the byte from the shift register is not copied into the SPIDR register until after the slave SPI has exited wait or stop mode. In slave mode, a received byte pending in the receive shift register is lost when entering wait or stop mode. An SPIF flag and SPIDR copy is generated only if wait mode is entered or exited during a transmission. If the slave enters wait mode in idle mode and exits wait mode in idle mode, neither a SPIF nor a SPIDR copy occurs.

SPI operation in wait mode depends upon the state of the SPISWAI bit in SPI control register 2.

- If SPISWAI is clear, the SPI operates normally when the CPU is in wait mode
- If SPISWAI is set, SPI clock generation ceases and the SPI module enters a power conservation state when the CPU is in wait mode.
- If SPISWAI is set and the SPI is configured for master, any transmission and reception in progress stops at wait mode entry. The transmission and reception resumes when the SPI exits wait mode.